

Exhibit 2

U.S. Patent No. 7,519,814 (“’814 Patent”)

Accused Instrumentalities: Amazon’s AWS End-of-Support Migration Program (“EMP”), and all versions and variations thereof since the issuance of the asserted patent.

Claim 1

Claim 1	Accused Instrumentalities
<p>[1pre] 1. In a system having a plurality of servers with operating systems that differ, operating in disparate computing environments, wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service, wherein the applications are executed in a secure environment, wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising:</p>	<p>To the extent the preamble is limiting, Amazon practices, through the Accused Instrumentalities, in a system having a plurality of servers with operating systems that differ, operating in disparate computing environments, wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service, wherein the applications are executed in a secure environment, wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, as claimed.</p> <p><i>See claim limitations below.</i></p> <p><i>See also, e.g.:</i></p> <p>AWS EMP for Windows Server enables application migration from legacy Windows Server systems to supported Windows operating systems. The EMP process includes an assessment of your application and testing requirements, packaging of the application and its dependencies on the legacy operating system based on your requirements, and migration to an AWS environment running a supported Windows Server version. When migrated to the new server environment, the application will run on the new OS as it would on the legacy operating system. The new package redirects application requests and does not include installation of any part of the legacy operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>

Claim 1	Accused Instrumentalities
	<p>EMP compatibility packages solve these challenges by resolving application dependencies on machine names, IP addresses, and ports that were created when the application was first installed on an out-of-support version of Windows Server. Network redirections are applied to packages so that the application can be migrated to a <u>new server</u>. EMP compatibility packages support Windows services common in Oracle Application Server, PeopleSoft, Tomcat, IBM WebSphere, and SQL Server 2005.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p> <p>AWS End-of-Support Migration Program (EMP) for Windows Server supports the following operating systems:</p> <ul style="list-style-type: none"> • Windows Server 2019 (64-bit) • Windows Server 2016 (64-bit) • Windows Server 2012 R2 (64-bit) • Windows Server 2008 R2 (64-bit) • Windows Server 2008 (32-bit and 64-bit) • Windows Server 2003 SP2 (32-bit) <p>The following system requirements must be met to migrate your application with AWS End-of-Support Migration Program (EMP) for Windows Server.</p> <ul style="list-style-type: none"> • .NET. Microsoft .NET 4.0 Client Profile, or later. • Memory. As required by the packaged applications. Minimum 2 GB. • Processor. As required by the packaged applications. Two CPUs recommended. • Disk space. 10 GB required to create the snapshots. The size of the EMP package depends on the size of the application. It will be 10 to 50 percent larger than the application being packaged. <p>https://docs.aws.amazon.com/emp/latest/userguide/emp-supported-os.html, Last accessed on July 13th, 2023</p> <p>Kernel-mode drivers that are a different bitness than the target operating system. Device drivers are not virtualized with EMP and therefore must be compatible with the target operating system. Compatible drivers can be deployed with the package. For example, if you are moving to a 64-bit operating system, you must have a 64-bit driver that is compatible with the new operating system.</p> <p>https://docs.aws.amazon.com/emp/latest/userguide/emp-limitations.html, Last accessed on July 13th, 2023</p>

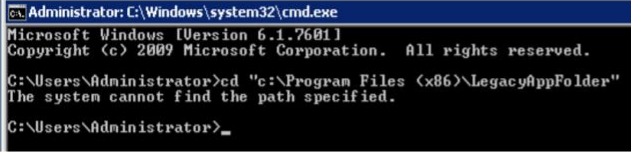
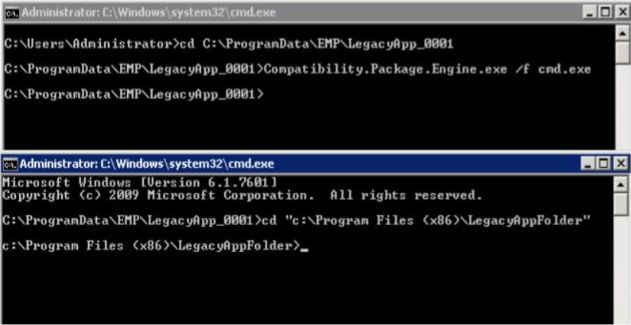
Claim 1	Accused Instrumentalities
	<p>Phase 2: Compatibility Packaging</p> <p>AWS SMEs utilize the EMP Compatibility Packager, along with the information provided in Phase 1, to identify and package applications with libraries, files, and other dependencies. The resulting package resolves OS dependencies and enables the application to successfully run on new versions of Windows Server.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p> <p>4.1.1 Application and Runtime Isolation</p> <p>Applications with conflicting requirements or outdated run times can safely run by being isolated from other applications on the server. Isolation enables EMP compatibility packages to include runtimes that conflict with other runtimes on the server, so that they are used only by the packaged application. Examples of conflicting requirements might include use of Java 1.3, .Net 2.0, and msxml.dll.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p> <p>Package an IIS-based application</p> <p>EMP supports packaging and migrating legacy IIS-based applications that run on Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 to the latest, supported versions of Windows Server running on AWS.</p> <p>https://docs.aws.amazon.com/pdfs/emp/latest/userguide/service-guide.pdf Last accessed on June 2nd, 2023</p> <p>Once the package has been launched by the start of a service, when an application shortcut or other entry point into the package is launched, another log file is created within the deployed package root folder called RunWorkflowLog.txt.</p> <p>https://docs.aws.amazon.com/pdfs/emp/latest/userguide/service-guide.pdf Last accessed on June 2nd, 2023</p>

Claim 1	Accused Instrumentalities
	<p>The package additionally includes a redirection engine that intercepts the API calls that the application makes to the underlying Windows Server OS, and redirects them to the files and registry within the created package. Therefore, the application always requests resources from within the package rather than from the new Windows Server OS, so that the application runs successfully on the new OS.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>
<p>[1a] storing in memory accessible to at least some of the servers a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers;</p>	<p>The method practiced by Amazon through the Accused Instrumentalities includes a step of storing in memory accessible to at least some of the servers a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers.</p> <p><i>See, e.g.:</i></p> <p>AWS EMP for Windows Server enables application migration from legacy Windows Server systems to supported Windows operating systems. The EMP process includes an assessment of your application and testing requirements, packaging of the application and its dependencies on the legacy operating system based on your requirements, and migration to an AWS environment running a supported Windows Server version. When migrated to the new server environment, the application will run on the new OS as it would on the legacy operating system. The new package redirects application requests and does not include installation of any part of the legacy operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p> <p>The application accesses data stored in a fixed location that is not available on the new OS version: the EMP engine redirects these requests to the appropriate location on the new version of the operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p>

Claim 1	Accused Instrumentalities
	<p>The resulting package includes everything that the application needs to run on a modern operating system, including application files, runtimes, components, and deployment tools. The package does not include the legacy operating system, which means you never run any part of the legacy Windows Server version on the new Windows Server to which the application is upgraded.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p> <p>The configuration of a package is defined in a series of XML files, which includes:</p> <ul style="list-style-type: none"> • Configurations for Registry keys and values (AppRegistry.xml) • File, Registry, and Network redirections (Redirections.xml) • File Type Associations (FileAssociations.xml) • Shortcuts (Shortcuts.xml) • Executable programs (Programs.xml) • Environment Variables (EnvironmentVariables.xml) <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p> <p>The package additionally includes a redirection engine that intercepts the API calls that the application makes to the underlying Windows Server OS, and redirects them to the files and registry within the created package. Therefore, the application always requests resources from within the package rather than from the new Windows Server OS, so that the application runs successfully on the new OS.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>

Claim 1	Accused Instrumentalities
	<p>Kernel-mode drivers that are a different bitness than the target operating system. Device drivers are not virtualized with EMP and therefore must be compatible with the target operating system. Compatible drivers can be deployed with the package. For example, if you are moving to a 64-bit operating system, you must have a 64-bit driver that is compatible with the new operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p>
<p>[1b] wherein the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems,</p>	<p>In the method practiced by Amazon through the Accused Instrumentalities, the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems.</p> <p><i>See, e.g.:</i></p> <p>AWS EMP for Windows Server enables application migration from legacy Windows Server systems to supported Windows operating systems. The EMP process includes an assessment of your application and testing requirements, packaging of the application and its dependencies on the legacy operating system based on your requirements, and migration to an AWS environment running a supported Windows Server version. When migrated to the new server environment, the application will run on the new OS as it would on the legacy operating system. The new package redirects application requests and does not include installation of any part of the legacy operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p> <p>The package additionally includes a redirection engine that intercepts the API calls that the application makes to the underlying Windows Server OS, and redirects them to the files and registry within the created package. Therefore, the application always requests resources from within the package rather than from the new Windows Server OS, so that the application runs successfully on the new OS.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p>

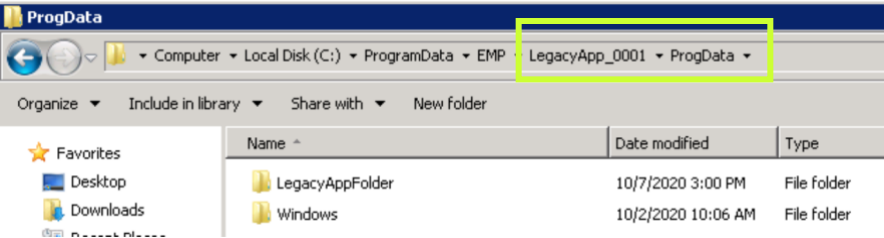
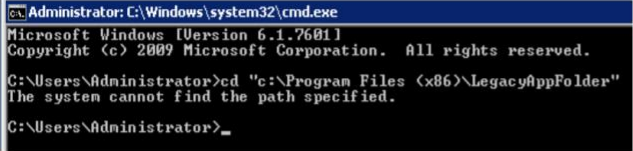
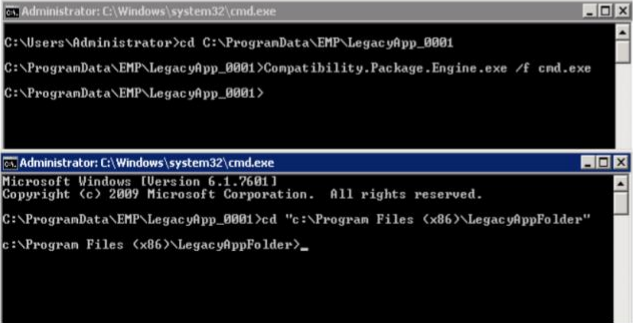
Claim 1	Accused Instrumentalities
	<p>Migrating legacy applications to newer, supported versions of Windows Server often requires refactoring, which is expensive and complex because of limited in-house expertise for such applications. With EMP for Windows Server, you no longer need to refactor your applications to ensure <u>compatibility with newer versions of Windows Server</u>. You now have the choice of using the tool as a self-service option at no cost or you can pay for the expert engagement to drive migration of your applications to AWS using the EMP tool.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p>
<p>[1c] the containers of application software excluding a kernel,</p>	<p>In the method practiced by Amazon through the Accused Instrumentalities, the containers of application software exclude a kernel.</p> <p><i>See, e.g.:</i></p> <p>4.3.1 Application to Operating System Compatibility</p> <p>EMP software runs legacy applications within a compatibility package on modern, up-to-date operating systems. The EMP compatibility package <u>does not contain any parts of the legacy operating system</u>. Instead, it intercepts the operating system requests and redirects and <u>resolves them against the up-to-date host operating environment</u>.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023 (annotated)</p>
<p>[1d] wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server,</p>	<p>In the method practiced by Amazon through the Accused Instrumentalities, some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server.</p> <p><i>See, e.g.:</i></p> <p>The package additionally includes a redirection engine that intercepts the API calls that the application makes to the underlying Windows Server OS, and redirects them to the files and registry within the created package. Therefore, the application always requests resources from within the package rather than from the new Windows Server OS, so that the application runs successfully on the new OS.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>

Claim 1	Accused Instrumentalities
	<p>4.3.3 Services and Drivers</p> <p>EMP Compatibility packages support Windows Services out of the box. Drivers aren't captured in an EMP package but can be extracted and deployed locally using the EMP deployment script feature, as long as they are compatible with the target operating system.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p> <p>Use cmd.exe to check for the file in the local system. The system should not be able to find the file.</p>  <p>Run cmd.exe as a child process to the package engine. This should confirm that the LegacyAppFolder is available in the context of the EMP package.</p>  <p>https://docs.aws.amazon.com/emp/latest/userguide/emp-run-cmd-child.html, Last accessed on June 8th, 2023</p>
[1e] wherein said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local	<p>In the method practiced by Amazon through the Accused Instrumentalities, said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Instrumentalities
<p>system files that remain resident on the server,</p>	<p>x64 (Engine Binaries) — Contains the EMP runtime files for packages to be deployed on a 64-bit system. When packaging is performed on a 64-bit machine, the contents of this folder are automatically copied into the root folder of the EMP package during the package build. An EMP package that was built on a 64-bit machine cannot be run on a 32-bit machine.</p> <p>x86 (Engine Binaries) — Contains the EMP runtime files for packages to be deployed on a 32-bit system. When packaging is performed on a 32-bit machine, the contents of this folder are automatically copied into the root folder of the EMP package during the package build. An EMP package that was built on a 32-bit machine will run on a 64-bit machine, however, we recommend that you use the appropriate runtimes for the destination platform. This is automatically handled during the deployment process.</p> <p>https://docs.aws.amazon.com/pdfs/emp/latest/userguide/service-guide.pdf Last accessed on June 2nd, 2023</p> <p>Applications with conflicting requirements or outdated run times can safely run by being isolated from other applications on the server. Isolation enables EMP compatibility packages to include runtimes that conflict with other runtimes on the server, so that they are used only by the packaged application. Examples of conflicting requirements might include use of Java 1.3, .Net 2.0, and msxml.dll.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>
<p>[1f] and wherein the application software cannot be shared between the plurality of secure containers of application software,</p>	<p>In the method practiced by Amazon through the Accused Instrumentalities, the application software cannot be shared between the plurality of secure containers of application software.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Instrumentalities
	<p>EMP software runs legacy applications within a compatibility package on modern, up-to-date operating systems. The EMP compatibility package does not contain any parts of the legacy operating system. Instead, it intercepts the operating system requests and redirects and resolves them against the up-to-date host operating environment.</p> <p>The EMP compatibility package permits the legacy application to run alongside other applications and/or other versions of the same application. For example, multiple versions of Microsoft Office can run simultaneously on the target operating system, or two incompatible 32-bit applications can run together, isolated from each other by the EMP compatibility package.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p> <p>Applications with conflicting requirements or outdated run times can safely run by being isolated from other applications on the server. Isolation enables EMP compatibility packages to include runtimes that conflict with other runtimes on the server, so that they are used only by the packaged application. Examples of conflicting requirements might include use of Java 1.3, .Net 2.0, and msxml.dll.</p> <p>https://d1.awsstatic.com/windows/AWS_EMP_WindowsServer_Whitepaper_V2.1.pdf Last accessed on June 2nd, 2023</p>
[1g] and wherein each of the containers has a unique root file system that is different from an operating system's root file system.	<p>In the method practiced by Amazon through the Accused Instrumentalities, each of the containers has a unique root file system that is different from an operating system's root file system.</p> <p><i>See, e.g.:</i></p>

Claim 1	Accused Instrumentalities
	<p>This section describes the folders and files that are included in an EMP compatibility package. When the EMP compatibility packaging process is complete, the output of the package builder is called an EMP compatibility package. The package contains both the file and registry data of the packaged application, and the EMP binaries and configuration files that are required to deploy and run the packaged application.</p> <p>The EMP package, which is the product to deploy, is called the source package. The post-deployment package is called the deployed package. These packages are slightly different at the level of the root package folder. However, the packaged application content is the same in both packages.</p> <p>https://docs.aws.amazon.com/pdfs/emp/latest/userguide/service-guide.pdf Last accessed on June 2nd, 2023</p> <p>x64 (Engine Binaries) — Contains the EMP runtime files for packages to be deployed on a 64-bit system. When packaging is performed on a 64-bit machine, the contents of this folder are automatically copied into the root folder of the EMP package during the package build. An EMP package that was built on a 64-bit machine cannot be run on a 32-bit machine.</p> <p>x86 (Engine Binaries) — Contains the EMP runtime files for packages to be deployed on a 32-bit system. When packaging is performed on a 32-bit machine, the contents of this folder are automatically copied into the root folder of the EMP package during the package build. An EMP package that was built on a 32-bit machine will run on a 64-bit machine, however, we recommend that you use the appropriate runtimes for the destination platform. This is automatically handled during the deployment process.</p> <p>https://docs.aws.amazon.com/pdfs/emp/latest/userguide/service-guide.pdf Last accessed on June 2nd, 2023</p>

Claim 1	Accused Instrumentalities
	<div><p>https://docs.aws.amazon.com/emp/latest/userguide/emp-run-cmd-child.html, Last accessed on June 8th, 2023 (annotated)</p><p>Use cmd.exe to check for the file in the local system. The system should not be able to find the file.</p><p>Run cmd.exe as a child process to the package engine. This should confirm that the LegacyAppFolder is available in the context of the EMP package.</p><p>https://docs.aws.amazon.com/emp/latest/userguide/emp-run-cmd-child.html, Last accessed on June 8th, 2023</p></div>